

Variance reduction for steady-state simulation and sensitivity analysis of stochastic chemical systems

Andreas Miliias-Argeitis *

John Lygeros †

Mustafa Khammash ‡

January 21, 2014

Abstract

We address the problem of estimating steady-state quantities associated to systems of stochastic chemical kinetics. In most cases of interest these systems are analytically intractable, and one has to resort to computational methods to estimate stationary values of cost functions. In this work we consider a previously introduced variance reduction method and present an algorithm for its application in the context of stochastic chemical kinetics. Using two numerical examples, we test the efficiency of the method for the calculation of steady-state parametric sensitivities and evaluate its performance in comparison to other estimation methods.

1 Introduction

Knowledge of steady-state quantities related to an ergodic stochastic chemical system can provide useful insights into its properties. Moreover, steady-state values of cost functions are often easier to estimate with good accuracy compared to stationary distributions. When the system propensities are affine in the state, mean values of polynomial functions of the system state can be computed analytically, as the system of moments is closed. However, when non-polynomial functions are considered, or the system propensities are not affine, analytic calculations are no longer possible and the only solution left is simulation.

While moment closure methods [20] can be used to provide good approximations to system moments over a finite time interval, they commonly tend to diverge from the true solution over time, thus resulting in biased steady-state values. The solution presented in Ref. [18] works only when polynomial functions of the state are considered, and generalization to arbitrary functions is still very difficult. The Finite State Projection algorithm [13] can be alternatively employed to provide moment estimates with guaranteed accuracy bounds, however the number of states required to attain a certain accuracy makes the method applicable to small problems.

On the other hand, stochastic simulation [5] can always provide estimates for the stationary mean of any function of the state, however these estimates are inevitably noisy. Brute-force noise reduction can only be achieved at an increased computational cost, either by simulating longer trajectories or by running many trajectories in parallel.

Another possibility for reducing the noise in the estimated quantities is the application of a variance reduction technique [3], provided the added computational cost of the reduced variance estimator is significantly smaller than the gain in computer time. In this work we present the application of such a variance reduction technique [9, 8] to systems of stochastic chemical kinetics. The idea is based on so-called *shadow functions* and originated in the queueing systems simulation literature, a field where the range of analytically tractable systems in that field is much larger. We demonstrate how the same idea can be applied to steady-state simulation of stochastic chemical systems. We further test the capabilities of the reduced-variance estimators by performing parametric sensitivity calculations for two systems governed by nonlinear propensity functions.

The paper is organized as follows: in Sections II and III we define the steady-state estimation problem and define the naïve and shadow function estimators. In Sections IV and V we present one possible implementation of the variance reduction technique to stochastic chemical kinetics and its applicability

*Department of Biosystems Science and Engineering, ETH Zurich, 4058 Basel, Switzerland; andreas.miliias@bsse.ethz.ch

†Automatic Control Lab, ETH Zurich, Physikstrasse 3, 8092, Zurich, Switzerland; lygeros@control.ee.ethz.ch

‡Department of Biosystems Science and Engineering, ETH Zurich, 4058 Basel, Switzerland; mustafa.khammash@bsse.ethz.ch

to steady-state sensitivity analysis. The numerical examples in Section VI serve to demonstrate the effectiveness of the shadow function method in practice and assess its computational cost in comparison to naïve estimation. The conclusions of our study and some future research directions are finally summarized in Section VII.

2 Problem statement

2.1 Setup

Assume an irreducible positive recurrent Markov chain $X = \{X(t) : t \geq 0\}$ on a countable space \mathcal{S} . In the case of stochastic chemical kinetics, $\mathcal{S} \subseteq \mathbb{Z}_{\geq 0}^N$, where N is the number of chemical species in the system. The chain moves according to a finite set of available transitions $\{\zeta_r \in \mathcal{S}\}_{r=1}^R$, with a corresponding set of propensity functions $\{\lambda_r : \mathcal{S} \rightarrow \mathbb{R}\}_{r=1}^R$. The infinitesimal generator of X is the operator Q satisfying

$$(Qf)(x) = \sum_{r=1}^R \lambda_r(x)(f(x + \zeta_r) - f(x)) \quad (1)$$

for all $f : \mathcal{S} \rightarrow \mathbb{R}$ such that $|Qf(x)| < \infty \forall x \in \mathcal{S}$. The discreteness of \mathcal{S} allows us to enumerate its elements and think of Q as an infinite matrix $Q = (Q_{ij})$, $i, j \in \mathbb{N}$. Similarly, any function f on \mathcal{S} can be thought of as an infinite column vector, and distributions on \mathcal{S} can be defined as infinite row vectors.

2.2 Steady-state estimators

Let $f : \mathcal{S} \rightarrow \mathbb{R}$ be a π -integrable cost function associated with X . The ergodic theorem for Markov chains [14] ascertains that for any initial condition

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t f(X(s)) ds = \pi f := \alpha \quad \text{almost surely,}$$

where π is the unique invariant distribution of the system and α the steady-state mean value of f . Since the analytic calculation of α is possible only in very special cases, its estimation from simulation is usually the only possibility. The most straightforward estimator of α is

$$\alpha_1(t) = \frac{1}{t} \int_0^t f(X(s)) ds, \quad (2)$$

which is also strongly consistent [3].

Under some further general conditions on X , and f , we also know that

$$\sqrt{t}(\alpha_1(t) - \alpha) \Rightarrow \sigma_1 \mathcal{N}(0, 1), \quad (3)$$

as $t \rightarrow \infty$, where \Rightarrow denotes weak convergence, $\mathcal{N}(0, 1)$ is the standard normal distribution and σ_1^2 is called the *time average variance constant (TAVC)* for $\alpha_1(t)$ [3].

The TAVC can be expressed in terms of the integrated autocovariance function of the process $(f_c(X(t)) : t \geq 0)$, where $f_c(x) := f(x) - \alpha$, according to the formula [3]:

$$\sigma_1^2 = 2 \int_0^\infty \mathbb{E}_\pi[f_c(X(0))f_c(X(s))] ds. \quad (4)$$

An alternative expression for σ^2 can be derived from the functional Central Limit Theorem for continuous-time Markov chains [4]:

$$\sigma_1^2 = -2\langle Qg, g \rangle = -2 \int g(x) \cdot Qg(x) d\pi(x), \quad (5)$$

where g is a solution to the so-called *Poisson's equation* [2] (Note that solutions to the Poisson equation are unique up to an additive constant, i.e. if g is a solution, then $g' = g + c$, $c \in \mathbb{R}$ is also a solution [2]):

$$Qg = -f_c. \quad (6)$$

A more general class of estimators for α has the form

$$\alpha_2(t) = \frac{1}{t} \int_0^t (f + h)(X(s)) ds, \quad (7)$$

where $h : \mathcal{S} \rightarrow \mathbb{R}$ is chosen such that $t^{-1} \int_0^t h(X(s)) ds \rightarrow 0$ almost surely for all $x \in \mathcal{S}$ [9]. The function h offers an extra degree of freedom in the design of the estimator, which can be exploited to achieve variance reduction. In other words, h can be chosen such that the TAVC of α_2 , denoted by σ_2^2 , is smaller than σ_1^2 . The obvious choice $h^{opt} = \alpha - f$ is of course intractable, however it suggests that a function h with a zero steady-state mean that is approximately equal to $\alpha - f$ could also achieve variance reduction. Such functions would result in a process $h(X(\cdot))$ that behaves almost antithetically from $f(X(\cdot))$, thus making the variance of $(f + h)(X(\cdot))$ smaller than that of $f(X(\cdot))$ alone. In the steady-state simulation literature, a function $h : \mathcal{S} \rightarrow \mathbb{R}$ that satisfies $\pi h = 0$ is called a *shadow function* [8].

The problem then becomes the selection of an appropriate shadow function h , so that $\sigma_2^2 = c \cdot \sigma_1^2$, with $c < 1$. From (3) we see that a reduction of variance by a factor c implies that the variance of $\alpha_2(T)$ is equal to the variance of $\alpha_1(T/c)$. Assuming that the computational cost of both estimators is dominated by the cost of simulating the process X , c^{-1} can be used as an indicator of the efficiency of α_2 relative to α_1 .

The basic idea of the shadow function method of Ref. [8], outlined in the next section, is to obtain such an h by using analytical information from a second Markov chain that approximates the original one and is mathematically tractable. A second alternative solution of more general applicability will be described after presenting the method in more detail.

3 The shadow function method: main idea

The basic idea to the shadow function method is to consider candidate functions of the form

$$h = Qg, \quad (8)$$

where Q is the generator matrix of the Markov chain and g is any π -integrable function (so that the ergodic theorem holds for it as well). In this case, and under the assumption that $\pi(Qg) = (\pi Q)g$ (that holds under some not-too-stringent conditions on g [8]), Qg becomes a shadow function. We are then naturally led to consider the solution of the Poisson equation (6), which could provide us with the appropriate function g . Solving (6) is of course not possible, since the state space is countable and α is unknown. However, we can look for so-called *surrogate functions* that approximate this solution to build a better estimator.

Following the analysis from [8], we consider another Markov chain \tilde{X} evolving on a countable space $\tilde{\mathcal{S}}$, with stationary distribution $\tilde{\pi}$ and generator \tilde{Q} . We also assume a map $r : \mathcal{S} \rightarrow \tilde{\mathcal{S}}$ (not necessarily one-to-one) and a function \tilde{f} that is somehow closely related to the original cost function f . If \tilde{f} is $\tilde{\pi}$ -integrable, we further assume that we can compute the solution to the Poisson equation

$$\tilde{Q}\tilde{g} = \tilde{\pi}\tilde{f} - \tilde{f}, \quad (9)$$

through which we arrive at a surrogate function

$$g(x) = \tilde{g}(r(x)) \quad \forall x \in \mathcal{S}. \quad (10)$$

Summing up, the approach outlined above is based on the fact that if \tilde{X} is: 1) a relatively good approximation of X and 2) tractable analytically, then we can derive a surrogate function g and an estimator $\alpha_2(t)$ which is better than the original estimator in terms of TAVC (assuming that the extra calculation time needed for α_2 is not significant).

4 Practical implementation of the shadow function method in chemical kinetics

The shadow function method was originally developed for steady-state simulation of queueing systems, for which a wide range of known and tractable approximations exists. The solution of the approximating Poisson equation can thus be calculated explicitly in many cases, and the application of the method is straightforward. This is not the case for stochastic chemical kinetic systems, where explicit solutions are very hard or impossible to calculate. One thus has to resort to different types of approximation schemes, outlined below.

4.1 State-space truncation

The Markov chains we are interested in satisfy the following properties:

1. They have a finite number of bounded increments over each finite time interval
2. Each state leads to a finite number of states (i.e. for every i , $Q(i, j) \neq 0$ for finitely many j 's)

For such chains, an obvious idea for obtaining an approximating process is to consider a chain evolving on a finite truncation of \mathcal{S} (i.e. consider $\tilde{\mathcal{S}}$ to be a finite subset of \mathcal{S}). Actually, under quite weak assumptions and careful definition of \tilde{Q} , one can show that the invariant distribution of \tilde{X} on $\tilde{\mathcal{S}}$ approaches that of X as the truncation size grows [21]. This of course implies that $\tilde{\pi}\tilde{f}$ also approaches πf . In this case, the function r between the two state spaces can be intuitively defined to map every $s \in \mathcal{S} \cap \tilde{\mathcal{S}}$ to itself, and every $s \in \mathcal{S} \setminus \tilde{\mathcal{S}}$ to some $\tilde{s} \in \tilde{\mathcal{S}}$ (which may vary with s). In this way, $\tilde{f} = f|_{\tilde{\mathcal{S}}}$.

In order to arrive at a good approximation with this approach, one first has to study a few simulations of X , to determine a finite set that contains a good amount of its invariant mass and then perform the necessary calculation of the solution to the Poisson equation on $\tilde{\mathcal{S}}$. The size of this set is determined in practice as a trade-off between tractability and approximation accuracy. However, the applicability of this approach is in general very limited due to the fact that the required truncations grow exponentially with the system dimension. Another problem is that the approximation \tilde{g} of g (the solution to the original intractable Poisson equation) will be very poor for states $s \in \mathcal{S} \setminus \tilde{\mathcal{S}}$, because of the form of r , which projects states outside $\tilde{\mathcal{S}}$ back into the set. This implies that significant variance reduction will be hard to achieve (and in some cases variance may even increase), if the chain sample paths exit $\tilde{\mathcal{S}}$ too frequently during simulation.

4.2 Approximating solutions of the Poisson equation

Instead of searching for an approximating Markov process, one may try to approximate the solution of (6) directly, to arrive at a suitable shadow function h . This approach is also followed in Ref. [12], where the discrete-time steady-state simulation problem is considered. Given a set of functions $\{\psi_i : \mathcal{S} \rightarrow \mathbb{R}, i = 1, \dots, n\}$ ¹, one can define

$$\hat{g} = \sum_{i=1}^n \theta_i \psi_i = \psi \cdot \theta, \quad (11)$$

where $\psi = [\psi_1 \dots \psi_n]$ and $\theta \in \mathbb{R}^{n \times 1}$ is a vector of weights.

In principle one could then try to calculate the value of θ that minimizes the TAVC of α_2 . Using (5) and (11), this variance constant turns out to be (see Appendix A)

$$\sigma_2^2 = \sigma_1^2 - 2[\langle f_c, \psi\theta \rangle - \langle Q(\psi\theta), g \rangle + \langle Q(\psi\theta), \psi\theta \rangle], \quad (12)$$

where g solves (6). Thus, minimizing the TAVC of (7) requires knowledge of g , which is unavailable.

We thus have to resort to heuristic methods for obtaining a suboptimal estimate of θ , for example by determining the value of θ that minimizes

$$L(\theta) := \int (Q(\psi\theta) + f_c)^2 d\mu(x)$$

for some suitable measure μ . This is a linear least squares regression problem, which can be solved approximately by generating a set of training data $(f_c(x_1), Q\psi(x_1)), \dots, (f_c(x_m), Q\psi(x_m))$, x_1, \dots, x_m , with weights $\mu(x_1), \dots, \mu(x_m)$.

If we define the finite-sample version of ψ by

$$\Psi := \begin{bmatrix} \psi_1(x_1) & \psi_2(x_1) & \dots & \psi_n(x_1) \\ \psi_1(x_2) & \psi_2(x_2) & \dots & \psi_n(x_2) \\ \dots & \dots & \dots & \dots \\ \psi_1(x_m) & \psi_2(x_m) & \dots & \psi_n(x_m) \end{bmatrix}$$

and similarly set

$$F_c := [f_c(x_1) \quad f_c(x_2) \quad \dots \quad f_c(x_m)]^T,$$

¹Candidate functions ψ_i must satisfy a boundedness condition derived from a Foster-Lyapunov inequality. For more details, see Ref. [7] or Ch.8 of Ref. [12]. In the sequel we will assume that all the functions considered satisfy this property.

we can then calculate the matrix $\Psi_Q \in \mathbb{R}^{m \times n}$ corresponding to $Q\psi$ by using the explicitly known form of the Markov chain generator (1) and finally obtain

$$\theta^* = (\Psi_Q^T M \Psi_Q)^{-1} \Psi_Q^T M F_c, \quad (13)$$

where $M = \text{diag}(\mu(x_1), \dots, \mu(x_m))$, as the (weighted) least squares minimizer of $L(\theta)$.

4.3 Variance reduction algorithm using a shadow function

Putting together all the elements presented above, we summarize below the basic steps of the variance reduction algorithm implemented in this work:

- (1) Simulate a long path of the process X using any preferred version of the stochastic simulation algorithm [5].
- (2) Obtain a rough estimate of α from the simulated trajectory using α_1 .
- (3) Pick a set of functions ψ_i , $i = 1, \dots, n$ and approximate the solution g to the Poisson equation (6) by $\hat{g} = \psi \cdot \theta^*$ using the approach outlined above.
- (4) Evaluate $h = Qg$ along the simulated sample path.
- (5) Refine the estimate of α using α_2 .
- (6) Verify that variance reduction has been achieved.

The last step is necessary to ensure that the variance has not actually increased due to the use of a suboptimal weight vector θ , and it can be carried out quite straightforwardly using the method of batch means [3] and the simulated trajectory from Step 1. In all cases we have tested, Steps 2-6 do not contribute more than a few seconds to the computational cost of this algorithm, which implies that the main computational bottleneck still lies at Step 1.

4.3.1 Implementation issues

The estimate of θ^* obtained by weighted least squares is clearly suboptimal, however it may still yield a reduced-variance estimator. The choice of the weighting measure μ in the optimization problem above is completely free, and one could in principle try to optimize over both μ and θ for a given problem. In practice however, such an approach would increase computational cost of the reduced-variance estimator and possibly eliminate the benefit of variance reduction. To maintain estimator efficiency, one should thus consider a single (or a few) “generic” choices for μ , and preferably re-use the points generated at Step 1.

A reasonable choice of weighting measure would be π itself. The training set for regression would then consist of all distinct points visited by the process over the course of simulation in Step 1 (possibly after discarding the burn-in period), weighted according to the empirical distribution of the process. A more coarse approximation of π would be to use the same sample with all weights being equal. Yet another possibility consists of sampling from a uniform grid that is centered on the area containing the bulk of the invariant mass of the chain. This area can also be crudely determined from the sample of Step 1. All these approaches can achieve variance reduction, however the optimal choice remains problem-dependent. Given that the calculation of least squares estimates can be carried out very efficiently using linear algebraic techniques, it is highly advisable to test several alternatives for the problem at hand.

In Ch.11 of Ref. [12], the problem of selecting an optimal θ is overcome by introducing a least-squares temporal difference learning (LSTD) algorithm for the approximation of the value of θ that minimizes the variance of α_2 in the context of discrete-time chains. The same algorithm could in principle be applied to continuous-time chains using the embedded discrete-time Markov chain and carrying out the necessary modifications to the original algorithm, based on the results of Ref. [10]. While this solution is theoretically justified, it requires setting up and running an LSTD estimator in parallel with the simulated chain that will asymptotically converge to the optimal value of θ . Depending on the convergence properties of this estimator, the overall efficiency of the variance reduction scheme may be smaller than the efficiency achieved by using a sub-optimal value for θ , especially when several approximating functions ψ_i are considered.

Another degree of freedom in the design of shadow function estimators is the choice of the approximating set $\{\psi_i, i = 1, \dots, n\}$. Here, the probabilistic interpretation of Poisson’s equation may assist the

selection of approximating functions by providing some useful intuition: Assuming f is π -integrable and X ergodic, it holds that [2, 11]

$$g(x) = \mathbb{E}_x \left[\int_0^{\tau(x_0)} f_c(X_s) ds \right],$$

where $\tau(x_0)$ is the hitting time of some state x_0 (changing x_0 simply shifts $g(x)$ by a constant) and \mathbb{E}_x denotes expectation given $X(0) = x$. From this equation one may infer some general properties of g (e.g. monotonicity, oscillatory behavior etc.) based on the form of the propensity functions. The same formula can be used to provide some crude simulation-based estimates of $g(x)$, which can be also helpful for the selection of $\{\psi_i\}$. Finally, a Lyapunov-type analysis can be employed to infer the asymptotic behavior of g [7].

5 Steady-state parameter sensitivity

Chemical reaction systems typically depend on several kinetic parameters, and the calculation of the output sensitivity with respect to these parameters is an essential step in the analysis of a given model. While there are several powerful parameter sensitivity methods available today [19, 1], they are mostly appropriate for transient sensitivity analysis, as the variance of their estimates tends to grow with the simulation length. Indeed, it can be shown that the variance of sensitivity methods based on the so-called likelihood ratio [6] or the Girsanov transformation [16] grows linearly with time. On the other hand, the variance of estimators based on finite parametric perturbations can be shown to remain bounded under mild conditions on the propensity functions, provided the underlying process is ergodic. However, the stationary variance can be still quite large, which makes necessary the use of a variance reduction method, such as the one presented here. Besides providing reduced-variance estimates of various steady-state functions of the chain, the shadow function estimator can be also employed for sensitivity analysis using a finite difference scheme [3] and the Common Random Numbers (CRN) estimator [17].

More analytically, assuming that the propensity functions of X are of the form $\lambda(x, p)$, where p is a parameter of interest, the finite difference method aims to characterize the sensitivity of the steady-state value of a given function f to a small finite perturbation of δ of p around a nominal value p_0 . If δ is small enough, we expect that $(\alpha(p_0 + \delta) - \alpha(p_0))/\delta$ will be approximately equal to $\partial\alpha/\partial p$.

Finite difference-based sensitivity analysis using shadow functions can be simply carried out by generating process trajectories for the nominal and perturbed parameter values, and estimating $\partial\alpha/\partial p$ by $(\alpha_2(p_0 + \delta) - \alpha_2(p_0))/\delta$. As shown in Ref. [17], use of the same random number stream for the generation of both the nominal and perturbed trajectories can result in great variance decrease compared to using independent streams.

6 Numerical Examples

To demonstrate the efficiency of shadow function estimators, we next present two applications of the method to steady-state sensitivity estimation. We compare our finite difference scheme that uses common random numbers and the shadow function estimator to the method of Coupled Finite Differences (CFD) [1], which frequently outperforms finite-difference estimators based on common random numbers and the Random Time Change representation [1, 17].

All numerical examples were generated using custom-written Matlab scripts running on a 3.4 Ghz quad-core PC with 8 GB of RAM.

6.1 Stochastic focusing

As a first example, we consider the stochastic focusing model of [15], where an input signaling molecule S inhibits the production of another molecule R . Stochastic focusing arises due to the presence of stochastic fluctuations in S , that make the mean value of R more sensitive to changes S than predicted by the deterministic model of the system. The same system is treated in Ref. [22] using a more sophisticated method based on trajectory reweighting.

The system reactions are given below:



where $k(S) = k_r/(S + K_m)$. The parameters used are $k_d = 100$, $k_r = 900$ and $K_m = 0.9$, while k_s is varied between 200 and 900 to study the effect of varying $\alpha_S := \mathbb{E}_\pi[S]$ on $\alpha_R := \mathbb{E}_\pi[R]$. More specifically (and similarly to Ref. [22]), we want to calculate the gain

$$g = \frac{\partial \ln(\alpha_R)}{\partial \ln(\alpha_S)} = \frac{k_s}{\alpha_R} \frac{\partial \alpha_R}{\partial k_s}.$$

To this end we estimate $\partial \alpha_R / \partial k_s$ using finite differences with $\delta = 2 \cdot 10^{-2} k_s$ at several points between $k_s = 200$ and $k_s = 900$. Figure 1 shows the calculated confidence intervals for $|g|$ obtained by the Common Random Number (CRN) estimator, the CRN estimator in conjunction with a shadow function and the CFD method. For each value of k_s , a simulated sample path of length $T = 8000$ time units (t.u.) was used to generate 19 batches of length 400 t.u. each, while the first 400 t.u. were discarded as burn-in.

Shadow functions consisted of linear combinations of all monomials in two variables up to order three (that is, $\psi_i = S^j \cdot R^k$, with $0 < j + k \leq 3$), together with $^2 \log(S + 2)$. This set of ψ_i 's was selected manually and is definitely not the "optimal" choice. The training set used for regression consisted of all unique points visited by the process sample paths after a burn-in period. Two alternative weighting schemes were tested for each value of k_s : according to the first, all points were assigned equal weight ($M = I$), while in the second one the points were weighted according to the empirical distribution of the process, calculated using the simulated sample paths ($M \approx \text{diag}(\pi)$). Both schemes lead to variance reduction, and calculation of θ^* in each case can be performed very fast (~ 0.15 sec), given the small number of training points (~ 2000).

Post-processing of the trajectories for the evaluation of the shadow function over the different batches takes another 5 sec of CPU time. On the other hand, SSA simulation takes on average 40 sec, which demonstrates that the overhead associated with the shadow function usage is relatively small, while the computational savings in the estimation of α_R are significant, as Table 1 demonstrates. Finally, a CFD simulation of the same length requires 220 sec of CPU time on average, while achieving a smaller magnitude of variance reduction.

Table 1: Variance reduction in the estimation of g

k_s/k_d	2	2.5	3	3.5	4	4.5	5	5.5	6	7	9
CFD: (σ_1^2/σ_2^2)	5.0	5.3	4.0	5.6	21.0	6.5	16.7	25.3	6.9	22.2	16.6
CRN+SF: (σ_1^2/σ_2^2)	7464	2051	968	735	444	379	232	475	386	108	129

Before we leave this example, we should point out that application of the shadow function method to just the birth-and-death process of S results in tremendous variance reduction for $f(S) = S^n$ and $n \leq 3$. As an example, Table 2 shows the confidence intervals of uncentered moment estimates obtained with and without a shadow function for $k_s = 500$, $k_d = 100$, using a simulated trajectory of $T = 5000$ t.u. and 10 batches. We attribute this phenomenon to the fact that the chosen set of functions ψ_i can approximate the true solution to the Poisson equation very closely for $n \leq 3$. Of course, it is known that S has a Poisson stationary distribution, which makes the use of a moment estimator pointless in this case. However, this interesting observation provides some heuristic justification for using polynomial approximating functions ψ_i .

Table 2: Variance reduction for a Poisson stationary distribution

Moment	True value	α_1 C.I.	α_2 C.I.
$\mathbb{E}_\pi[S]$	5	5.0012 ± 0.008	$5 \pm 7.8 \cdot 10^{-15}$
$\mathbb{E}_\pi[S^2]$	30	30.0241 ± 0.0721	$30 \pm 1 \cdot 10^{-12}$
$\mathbb{E}_\pi[S^3]$	205	205.42 ± 1.219	$205 \pm 3.3 \cdot 10^{-11}$
$\mathbb{E}_\pi[S^4]$	1555	1561.5 ± 18.88	1555.5 ± 1.05

²This is an example where the probabilistic interpretation of the Poisson equation given in subsection 4.3.1 can provide useful intuition for the selection of approximating functions. In the case at hand, $f = R$, so g (the solution to the Poisson equation) is expected to grow only very slowly with S , as the production rate of R tends to zero as $S \rightarrow \infty$.

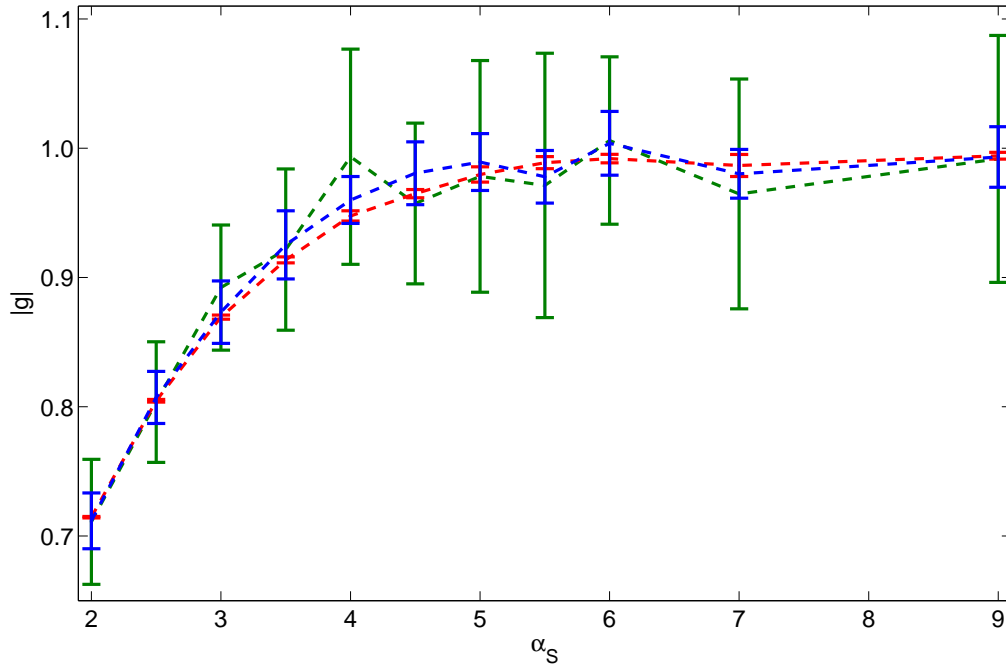


Figure 1: Absolute value of steady-state gain from α_S to α_R , estimated with the finite difference method. Shown are 95% confidence intervals obtained with the method of batch means [3]. Green: CRN estimator. Blue: CFD estimator. Red: CRN combined with shadow functions.

6.2 A six-dimensional system

Our second example is a system consisting of two interacting genes, A and B . The product of gene A forms homodimers, which repress the expression of A , as well as heterodimers with the product of B , which repress the expression of B . The system species are listed in Table 3, while Table 4 displays the reaction scheme and propensities of our model. Note that several parameters are assumed to be the same for the two genes for simplicity. For the same reason, the gene states are omitted from the model.

Table 3: Molecular species of the two-gene system

Name	Symbol
Gene A mRNA	m_A
Protein A monomer	p_A
Protein A dimer	p_{A_2}
Gene B mRNA	m_B
Protein B monomer	p_{B_2}
A-B dimer	p_{AB}

The system comprises six molecular species interacting through twelve reactions³. Our goal is to estimate the sensitivity of the steady-state mean of p_{AB} (the second repressor dimer), denoted by α_{AB} , to small variations of each of the system parameters. Once more, we compare the behavior of the CRN steady-state estimator with and without a shadow function to the performance of the CFD method. Shadow functions for this system consisted of linear combinations of all monomials of state pairs up to order two. Since the number of unique points visited by this six-dimensional process during simulation was (expectedly) too large to be handled with the least squares method, 10000 points sampled uniformly at random from this set were used in the regression step.

For the finite difference method we perturbed each parameter p by $\delta = 10^{-2} \cdot p$ and estimated the 95%

³Note that all examples presented in Ref. [22] consist of two-species systems and no more than four reactions

Table 4: Reactions and propensities

Reactions	Propensities
$\emptyset \xrightarrow{\lambda_1} m_A \xrightarrow{\lambda_2} \emptyset$	$\lambda_1 = k_r \frac{\phi^4}{\phi^4 + p_{A_2}^4}, \lambda_2 = k_{dr} m_A$
$\emptyset \xrightarrow{\lambda_3} p_A \xrightarrow{\lambda_4} \emptyset$	$\lambda_3 = k_p m_A, \lambda_4 = k_{dp} p_A$
$p_A + p_A \xrightleftharpoons[\lambda_6]{\lambda_5} p_{A_2}$	$\lambda_5 = k_1 p_A (p_A - 1), \lambda_6 = k_2 p_{A_2}$
$\emptyset \xrightarrow{\lambda_7} m_B \xrightarrow{\lambda_8} \emptyset$	$\lambda_1 = k_r \frac{\phi^2}{\phi^2 + p_{AB}^2}, \lambda_2 = k_{dr} m_B$
$\emptyset \xrightarrow{\lambda_9} p_B \xrightarrow{\lambda_{10}} \emptyset$	$\lambda_9 = k_p m_A, \lambda_{10} = k_{dp} p_A$
$p_A + p_B \xrightleftharpoons[\lambda_{12}]{\lambda_{11}} p_{AB}$	$\lambda_{11} = k_3 p_A p_B, \lambda_{12} = k_4 p_{AB}$

confidence intervals of each sensitivity estimate using batch means with 24 batches of length 4000 time units each (with an additional 4000 t.u. for burn-in). Prior to parameter perturbations, the estimate of α_{AB} was calculated for

$$\begin{aligned}
p_0 &= [k_r \quad \phi \quad k_{dr} \quad k_p \quad k_{dp} \quad k_1 \quad k_2 \quad k_3 \quad k_4] \\
&= [1 \quad 60 \quad 0.1 \quad 1 \quad 0.5 \quad 0.02 \quad 0.08 \quad 0.02 \quad 0.1].
\end{aligned}$$

The estimates and their corresponding variances were: $\alpha_1 = 64.46$, $\sigma_1^2 = 2.68$, $\alpha_2 = 65.31$ and $\sigma_2^2 = 3.5 \cdot 10^{-3}$. The results of the sensitivity analysis are summarized in Table 5. CRN sensitivity estimates and their associated confidence intervals not accurate enough to provide useful information. On the contrary, using a shadow function results in great improvements, as now the relative magnitudes and signs of the various sensitivity coefficients can be meaningfully compared to each other.

Table 5: Sensitivity estimates and associated confidence intervals for the two-gene system. Note that a reduction of a confidence interval by a factor r requires a variance reduction by a factor r^2 , which can be achieved by running simulations r^2 times longer.

Normalized sensitivity coefficient	CRN 95% CI	CFD 95% CI	CRN+SF 95% CI
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_r$	2.461 ± 1.339	1.048 ± 0.104	0.982 ± 0.034
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial \phi$	0.025 ± 0.020	0.013 ± 0.002	0.013 ± 0.0005
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_{dr}$	1.853 ± 11.222	-10.246 ± 1.453	-9.777 ± 0.232
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_p$	1.894 ± 1.236	1.020 ± 0.122	0.967 ± 0.025
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_{dp}$	0.513 ± 2.049	-1.876 ± 0.286	-1.888 ± 0.058
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_1$	137.769 ± 123.344	18.437 ± 8.181	22.374 ± 3.383
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_2$	21.675 ± 16.660	-3.053 ± 1.417	-2.785 ± 0.333
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_3$	87.001 ± 62.754	27.001 ± 4.43	23.351 ± 1.704
$\alpha_2^{-1} \cdot \partial \alpha_{AB} / \partial k_4$	8.003 ± 16.060	-4.818 ± 0.756	-4.597 ± 0.328

The variance reduction method remains quite efficient computationally in this case as well: SSA simulation of a 10^5 t.u. trajectory takes about 17 sec of CPU time, while calculation of θ requires 1 sec and post-processing of the sample path another 3 sec. At the same time, a CFD simulation of the same length requires 95 sec of CPU time on average, while failing to achieve a comparable level of variance reduction.

7 Discussion

We demonstrated the applicability of the powerful shadow function method to the problem of steady-state simulation of stochastic chemical kinetics. Our results suggest that a significant increase in the efficiency of a steady-state estimator is possible by only a small increase in its computational cost. The method can

be applied to the steady-state estimation of practically any function of the process, and can thus provide improved estimates of high order (cross-)moments, as well as estimates of stationary probabilities for subsets of the process state space, by using set indicators as cost functions. The magnitude of variance reduction achieved by the shadow function method allows also the efficient and precise computation of steady-state parameter sensitivities using the finite difference method.

The comparison of the efficiency of this approach for providing steady-state sensitivity estimates with the one presented in [22] is the topic of our ongoing work. It would also be instructive to assess the relative strengths and weaknesses of the LSTD approximation algorithm for optimizing the shadow function [12] and test its scalability with system size and number of approximating functions (note that only one-dimensional examples are treated in [12]).

The proposed workflow for arriving at a useful shadow function can be improved at several points, by drawing from the large literature on function approximation techniques, in order to enlarge its range of applicability and its accuracy. However, even a crude approach such as the one presented above seems to be sufficient for systems of practical interest.

A TAVC for shadow function estimator

From (5) and (11), $\sigma_2^2 = -2\langle Qg_2, g_2 \rangle$, where g_2 solves the Poisson equation $Qg_2 = -f_c - Q(\psi\theta)$. This implies that $g_2 = g_1 - \psi\theta$, where g_1 is the solution of the Poisson equation $Qg_1 = -f_c$. The variance of the shadow function estimator thus becomes

$$\begin{aligned}\sigma_2^2 &= -2\langle -f_c - Q(\psi\theta), g_1 - \psi\theta \rangle \\ &= -2[\langle -f_c, g_1 \rangle + \langle f_c, \psi\theta \rangle - \langle Q(\psi\theta), g_1 \rangle + \langle Q(\psi\theta), \psi\theta \rangle] \\ &= \sigma_1^2 - 2[\langle f_c, \psi\theta \rangle - \langle Q(\psi\theta), g_1 \rangle + \langle Q(\psi\theta), \psi\theta \rangle].\end{aligned}$$

References

- [1] ANDERSON, D. F. An efficient finite difference method for parameter sensitivities of continuous time markov chains. *SIAM Journal on Numerical Analysis* 50, 5 (2012), 2237–2258.
- [2] ASMUSSEN, S. *Applied probability and queues*. Springer, 2003.
- [3] ASMUSSEN, S., AND GLYNN, P. *Stochastic simulation: Algorithms and analysis*, vol. 57. Springer, 2007.
- [4] BHATTACHARYA, R. N. On the functional central limit theorem and the law of the iterated logarithm for markov processes. *Probability Theory and Related Fields* 60, 2 (1982), 185–201.
- [5] GILLESPIE, D. T. Stochastic simulation of chemical kinetics. *Annu. Rev. Phys. Chem.* 58 (2007), 35–55.
- [6] GLYNN, P. W. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM* 33, 10 (1990), 75–84.
- [7] GLYNN, P. W., AND MEYN, S. P. A Lyapunov bound for solutions of the Poisson equation. *The Annals of Probability* 24, 2 (1996), 916–931.
- [8] HENDERSON, S. *Variance Reduction Via an Approximating Markov Process*. PhD thesis, Dept. of Operations Research, Stanford University, 1997.
- [9] HENDERSON, S., AND GLYNN, P. Approximating martingales for variance reduction in markov process simulation. *Mathematics of Operations Research* 27, 2 (2002), 253–271.
- [10] HORDIJK, A., IGLEHART, D. L., AND SCHAASBERGER, R. Discrete time methods for simulating continuous time markov chains. *Advances in Applied Probability* (1976), 772–788.
- [11] MAKOWSKI, A. M., AND SHWARTZ, A. The poisson equation for countable markov chains: probabilistic methods and interpretations. In *Handbook of Markov decision processes*. Springer, 2002, pp. 269–303.
- [12] MEYN, S. *Control techniques for complex networks*. Cambridge University Press, 2007.

- [13] MUNSKY, B., AND KHAMMASH, M. The finite state projection algorithm for the solution of the chemical master equation. *The Journal of Chemical Physics* 124 (2006), 044104.
- [14] NORRIS, J. R. *Markov chains*. Cambridge University Press, 1998.
- [15] PAULSSON, J., BERG, O. G., AND EHRENBERG, M. Stochastic focusing: Fluctuation-enhanced sensitivity of intracellular regulation. *Proceedings of the National Academy of Sciences* 97, 13 (2000), 7148–7153.
- [16] PLYASUNOV, S., AND ARKIN, A. P. Efficient stochastic sensitivity analysis of discrete event systems. *Journal of Computational Physics* 221, 2 (2007), 724–738.
- [17] RATHINAM, M., SHEPPARD, P. W., AND KHAMMASH, M. Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks. *The Journal of chemical physics* 132, 3 (2010).
- [18] RUESS, J., MILIAS-ARGEITIS, A., SUMMERS, S., AND LYGEROS, J. Moment estimation for chemically reacting systems by extended kalman filtering. *The Journal of chemical physics* 135, 16 (2011), 165102–165102.
- [19] SHEPPARD, P. W., RATHINAM, M., AND KHAMMASH, M. A pathwise derivative approach to the computation of parameter sensitivities in discrete stochastic chemical systems. *The Journal of chemical physics* 136 (2012), 034115.
- [20] SINGH, A., AND HESPANHA, J. P. Approximate moment dynamics for chemically reacting systems. *Automatic Control, IEEE Transactions on* 56, 2 (2011), 414–418.
- [21] TWEEDIE, R. L. Truncation approximations of invariant measures for Markov chains. *Journal of applied probability* (1998), 517–536.
- [22] WARREN, P. B., AND ALLEN, R. J. Steady-state parameter sensitivity in stochastic modeling via trajectory reweighting. *The Journal of Chemical Physics* 136 (2012), 104106.